

REMARKS

Introductory Comments

This Amendment is submitted in response to the outstanding Office Action wherein the Examiner finally rejected claims 1 and 3-15, the only claims under consideration in the above-identified application. Applicants wish to thank Examiner Hoang for the time and courtesy shown to Applicants' attorney during the telephonic interview of December 4, 2007. Applicants make these amendments in light of the discussion during the interview to place the claims in condition for allowance or, in the alternative, in better condition for appeal.

By this Amendment, Applicants have amended independent claims 1 and 8 to include elements recited in dependent claim 9, which dependent claim has been canceled herein without prejudice or disclaimer. Dependent claims 3, 4, 7, 10, 12 and 13 are amended to provide proper antecedent basis and provide further clarification. These amendments do not introduce any new matter into the application. Consequently, upon entry of this Amendment, claims 1, 3-8, and 10-15 are pending in the application.

Entry and consideration of this Amendment are proper under 37 C.F.R. § 1.116 for at least the following reasons. This Amendment does not raise new issues requiring further search or consideration. For example, the elements added to independent claims 1 and 8 have been substantively examined as dependent claim 9 and do not introduce any new issues. Therefore, entry of this Amendment is proper under 37 C.F.R. § 1.116 and is hereby requested.

Applicants respectfully request reconsideration of the presently pending claims in light of the amendments above and the remarks that follow. Further, Applicants believe that there are also reasons, other than those set forth below, that explain why the pending claims are patentable. Applicants hereby reserve the right to set forth these reasons, and to argue for the patentability of claims not explicitly addressed herein, in future papers.

Amendment of Specification

In the outstanding Office Action, the Examiner objected to the specification due to the use of a certain trademark. Applicants have amended the specification to appropriately refer to trademarks referenced therein. Accordingly, it is respectfully requested that the objection to the specification be withdrawn.

Rejection of Claims 1-15 under 35 U.S.C. § 101

Claims 1-15 were rejected under 35 U.S.C. §101 (“Section 101”) as being directed to non-statutory subject matter, in particular a computing architecture comprising software per se. Office Action, page 3. The Examiner states that the body of claim 1 does not comprise any hardware to execute the computing machine claimed in the preamble. Office Action, page 7.

Applicant respectfully disagrees at least because claim 1 is directed to “a computing machine having a computing architecture.” Emphasis added. Nevertheless, to expedite prosecution of this application, Applicants have amended the body of independent claim 1 to further recite: “a base operating system (OS) installed to the computing machine,” and wherein “the computing machine is configured such that” The amendment made in this regard to claim 1 does not, and is not intended to, narrow or alter the scope of the claim in any way. Rather, the amendments are made solely to clarify the subject matter of the claims with regard to Section 101 as requested by the Examiner.

Claim 2 was previously canceled in Applicants’ Amendment dated July 9, 2007 and so its rejection is deemed moot. Claims 3-7 depend from independent claim 1 and refer to the “computing machine of claim 1.” It is believed that these amendments address the Examiner’s concerns. Therefore, the Section 101 rejection of claims 1-7 should be reconsidered and withdrawn.

Claim 8, and claims 10-15, which are dependent directly or indirectly therefrom, are

directed to a “[a] method of configuring a computer” Emphasis added. For at least these reasons, the Section 101 rejection of claim 8 and its dependent claims (claims 10-15) should be reconsidered and withdrawn as claim 8 is directed to patentable subject matter.

Rejection of Claims 1- 3, 5, 6, and 8 under 35 U.S.C. § 102(b)

Claims 1-3, 5, 6, and 8 were rejected under 35 U.S.C. §102(b) (“Section 102”) as being anticipated by Hall et al., “A Virtual Operating System,” pages 495-502 (“Hall”). For at least the reasons discussed below, Applicant respectfully traverses the Section 102 rejection of claims 1- 3, 5, 6, and 8.

At the outset, Applicant wishes to point out that “[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). See M.P.E.P. § 2131.

Further, “[t]o establish inherency, the extrinsic evidence ‘must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.’” *In re Robertson*, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (citations omitted). “[T]he examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art.” *Ex parte Levy*, 17 USPQ2d 1461, 1464 (BPAI 1990) (emphasis in original); see also, MPEP § 2112 (quoting *Levy*).

- A. “[T]he virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry”

The invention is defined by independent claims 1 and 8. These claims, as amended, recite:

1. A computing machine having a computing architecture, comprising:
a base operating system (OS) installed to the computing machine, the base OS having a base OS file system and a base OS registry;
at least one virtual OS environment within the base OS, the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry;
wherein the computing machine is configured such that attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry.
8. A method of configuring a computer with a base operating system (OS) having a base OS file system and registry, the method comprising the steps of:
creating at least one virtual OS environment under the base OS, each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry;
configuring the computer such that attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry.

Referring now to the reference cited by the Examiner, Hall attempts to solve a problem that existed more than 25 years ago, namely, organizations being hesitant to move to new hardware systems (with their associated operating systems) because of the high cost of having to train personnel on the new systems and porting software over to such systems. Hall proposed a “uniform system interface” that may be run on top of various operating systems and called this interface a “virtual operating system.” Hall, page 495, col. 1 and col. 3. According to Hall, this is effective because with computer users “there is no need to distinguish between the interface to an operating system and the operating system itself.” Hall, page 495, col. 3. Fig. 1 of Hall further shows how this virtual machine (i.e. system interface) consists of “interfacing the standardized virtual machine to the vendor supplied system.” In essence, this virtual machine (i.e. system interface) is simply a wrapper on top of a base operating system.

Hall does not disclose “the virtual OS environment having a virtual OS file system and a

virtual OS registry which are independent of the base OS file system and the base OS registry” as is affirmatively recited in claim 1 or “each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8.

In rejecting these claim elements, the Examiner alleges that Hall, at page 497, col. 1, section 4, discloses “file systems” and that a “registry” is inherent for an operating system. Office Action, page 3. However, even if these assertions are assumed to be accurate for sake of argument, a mere disclosure of “file systems” in Hall and a registry being inherent for an OS, taken alone or together, do not amount to a disclosure of “the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry” as is recited in claim 1 or “each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8. Emphasis added.

The cited section of Hall states:

To test the approach, a uniform program development environment was installed on several distinct systems. A program development environment consists of resources which assist programmers in the development and maintenance of computer programs, such as text editors, programming language processors, and file systems. The types of system resources with which such a virtual machine is concerned (files, directories, processes, and the user environment) require a general-purpose operating system interface.

Hall, page 497, col. 1, section 4. The reference to “file systems” in this section of Hall does not include any teaching or suggestion that the “file systems” are part of a virtual OS environment or that the file systems are independent of a base OS file system. Rather, the reference merely describes resources (e.g., a text editor) that are available in a program development environment to assist programmers in the development and maintenance of computer programs. Moreover, the program development environment in the cited section of Hall is used merely to create source

code for a virtual operating system, and there is no teaching or suggestion in Hall of the virtual operation system having a file system that is independent of the base OS file system.

The Examiner also cited the following section of Hall:

In all cases, the system was offered in parallel with the existing environment, thereby allowing users to experiment with the virtual operating system without giving up the familiar, vendor-supplied environment.

Hall, page 497, col. 3. Emphasis added. The Examiner uses this citation in an attempt to show the virtual operating system of Hall as being independent of the base OS file system and the base OS registry. However, Applicants respectfully disagree with this argument. The reference to “parallel” does not mean, or even imply, that the virtual operating system (i.e. system interface) is “independent of the base OS file system and the base OS registry” as is asserted by the Examiner. It is Applicants’ understanding and belief that the term “parallel,” means simultaneously or concurrently. In other words, the virtual operating system (i.e. system interface) may be run simultaneously or concurrently with the “familiar, vendor-supplied” interface of the base operating system.

In fact, as discussed above, Hall teaches away from “the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry” as is recited in claim 1 or “each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8. Emphasis added. For instance, the disclosed purpose of the virtual operating system in Hall is to provide a standard interface to a real operating system. Hall, page 496, Fig. 1 and col. 1, first and second paragraphs. In other words, the virtual operating system in Hall is simply a wrapper on top of a real operating system, i.e. an interface between a user interface and a real operating system. Hall states, “the emphasis in building a virtual operating system is on the interface presented to the user.” Hall, page 496, col. 1, fourth paragraph. As a mere interface on

top of a real operating system, there is no need for the virtual operating system of Hall to have a file system or a registry that is independent of the file system or registry of the real operating system. In fact, by focusing on providing a standardized interface to a real operating system, Hall teaches away from such a configuration. Hall, page 495, col. 3, last paragraph.

Moreover, even if a registry is assumed, *arguendo*, to be inherent for an operating system as asserted in the Office Action, this does not amount to a teaching, explicit or inherent, of a virtual OS environment having a virtual OS registry that is independent of the base OS registry. The virtual operating system of Hall, as a mere interface to a real operating system, teaches away from an independent registry for the virtual operating system.

For at least these reasons, Hall fails to disclose “the virtual OS environment having a virtual OS file system and a virtual OS registry which are independent of the base OS file system and the base OS registry” as is recited in claim 1 or “each virtual OS environment having a virtual file system and registry which are independent of the base OS file system and registry” as is recited in claim 8. Emphasis added.

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. Therefore, the Section 102 rejection of independent claims 1 and 8 should be withdrawn and the claims allowed.

B. “[A]t least one virtual OS environment within the base OS”

Hall also fails to disclose other elements recited in independent claims 1 and 8. For example, Hall fails to disclose “at least one virtual OS environment within the base OS” as recited in claim 1 and “creating at least one virtual OS environment under the base OS” as recited in claim 8. Emphasis added. The Office Action fails to point to any particular teaching in Hall of a virtual OS environment that is within or under a base OS. Moreover, Hall teaches away from these claim elements. For example, Fig. 1 in Hall (page 496 of Hall) shows that the virtual

operating system of Hall, as an interface to a real operating system, is on top of a vendor supplied system (i.e., a wrapper for a real operating system). Therefore, Hall fails to disclose “at least one virtual OS environment within the base OS” as recited in claim 1 and “creating at least one virtual OS environment under the base OS” as recited in claim 8. Emphasis added. For this independent reason, the Section 102 rejection of independent claims 1 and 8 should be withdrawn and the claims allowed.

C. “[A]ttempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry”

Further, Hall fails to disclose other elements recited in independent claims 1 and 8. For example, Hall fails to disclose “attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry” as recited in claim 1 and “attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry” as recited in claim 8. Emphasis added. Applicant can find no reference within Hall that teaches, enables, or suggests the redirection of calls made to the base OS by an application running under the virtual OS environment.

Furthermore, U.S. Patent No. 6,141,698 to Krishnan et al. (“Krishnan”), does not teach or suggest “attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry” as recited in claim 1 and “attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry” as recited in claim 8. In rejecting dependent claim 9 (which elements are now affirmatively recited in independent claims 1 and 8), the

Examiner relied on Krishnan to teach injecting DLLs and points to the Abstract and Figures 2, 3, and 6 of Krishnan.

Upon a review of Krishnan, Krishnan is directed to a:

[M]ethod and system for modifying the behavior of existing executable code by injecting new code into an executable file. The injection mechanism injects a reference to new code contained in a DLL into an existing executable file such that, when the code of the executable file is executed, the DLL is automatically loaded and the new code is automatically executed. A reference to the DLL is injected into the executable file by either modifying an import table of the file, which causes automatic loading of the DLLs referred to therein, or by adding DLL loader code to the file.

Krishnan, Abstract. Krishnan teaches that injecting code into an executable file may be helpful in situations such as when a third party vendor, who does not have access to application source code, may wish to incorporate vendor-specific code into the application before redistributing to an end customer. Krishnan, col. 1, lines 23-30. As another example, Krishnan teaches that injecting new code into existing application code is helpful in preventing software pirates from unencrypting application programs and making illegal copies of such application programs. Krishnan, col. 1, line 31 – col. 2, line 28.

However, a simple modification of existing application code using a DLL as disclosed in Krishnan is entirely different from the recitations in Applicants' claim 1 and claim 8. Clearly, Krishnan does not teach or even contemplate "attempts to access the base OS file system and the base OS registry by an application running under the virtual OS environment are redirected to the virtual OS file system and the virtual OS registry" as recited in claim 1 and "attempts to access the base OS file system and registry by at least one application running under the virtual OS environment are redirected to the virtual OS environment file system and registry" as recited in claim 8. Should the Examiner continue to rely on Krishnan to reject these claim elements, Applicants respectfully request that the Examiner specifically explain how the injection of a DLL into existing code could possibly teach or suggest same.

D. Dependent Claims 2, 3, 5, and 6

In view of the above, it is respectfully contended that Hall fails to disclose, expressly or under the principles of inherency, each and every element of the invention as defined by independent claims 1 and 8. In addition, for at least the same reasons presented above, the Section 102 rejection of dependent claims 3, 5, and 6 should be withdrawn and the claims allowed because the claims depend from independent claim 1. Claim 2 was previously canceled in Applicants' Amendment dated July 9, 2007 and so its rejection is deemed moot. It is therefore respectfully requested that the rejection of claims 1-3, 5, 6, and 8 under Section 102 based on Hall be withdrawn.

Rejection of Claims 4, 7, and 9-15 under 35 U.S.C. § 103(a)

Claims 4, 7, and 9-11 were rejected under 35 U.S.C. § 103(a) ("Section 103") as being unpatentable over Hall in view of Krishnan. Applicant believes the Office Action to include a typographical error and that the Examiner intended to reject all remaining claims, namely claims 4, 7, 9-15 over Hall in view of Krishnan.

By this Amendment, Applicants have amended independent claims 1 and 8 to include elements recited in dependent claim 9, and have canceled claim 9 without prejudice or disclaimer. Accordingly, the rejection of claim 9 is hereby obviated.

Furthermore, as discussed above, Krishnan does not cure the deficiencies of Hall with respect to independent claims 1 and 8. Therefore, neither Hall nor Krishnan, either alone or in combination, teach, disclose or suggest all claim elements recited in claims 1 and 8. Therefore, the Section 103 rejection of claims 4, 7, and 10-15 should be withdrawn and the claims allowed based at least on their dependencies from claim 1 or 8.

Moreover, claims 4, 7, and 10-15 also recite independently patentable subject matter. For example, claim 12 recites the step of "creating a copy of the base OS file system and registry in

the virtual OS environment file system and registry.” The Examiner asserts that Krishnan teaches this feature at column 4, line 55 – column 5, line 5. Office Action, page 6.

The section of Krishnan cited by the Examiner states:

The injection mechanism is useful in many scenarios. For example, in a globally networked system such as the Internet, licensing code can be incorporated into an existing application and distributed on the system by injecting the licensing code into the application using the injection mechanism. The licensing developer creates a new DLL with the new licensing code accessible through the initialization function of the DLL. The developer then uses the injection mechanism of the present invention to create a modified version of the application that includes a reference to the new DLL. This modified version is then distributed. Further, the newly injected licensing code can be made more secure by using the injection mechanism to inject security code into the modified application. The injected security code makes it impossible to recreate in a reasonable amount of time an unmodified version of the application that does not include the injected licensing DLL.

Applicant respectfully submits it does not understand how this Krishnan citation (or Krishnan or Hall generally) teaches or suggests the step of “creating a copy of the base OS file system and registry in the virtual OS environment file system and registry” as is affirmatively recited in claim 12. Should the Examiner continue to rely upon Krishnan to reject this claim, Applicants respectfully request that the Examiner explain in more detail the Examiner’s rationale so that Applicants may provide an appropriate response.

Claim 13 recites “wherein an application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry.” The Examiner rejects this claim under the same argument used for claim 12 including relying on the same cited portion of Krishnan. Again, Applicant respectfully submits it does not understand how this Krishnan citation (or Krishnan or Hall generally) teaches or suggests “wherein an application running under the virtual OS environment is executed using the copy in the virtual OS environment file system and registry” as is affirmatively recited in claim 13. Should the Examiner continue to rely upon Krishnan to reject this claim, Applicants respectfully request that

the Examiner explain in more detail the Examiner's rationale so that Applicants may provide an appropriate response.

Claim 14 requires "setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory."

According to the Examiner, claim 14:

[W]ould have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that the injecting DLL method and techniques can be applied to different computing environment because the injecting DLL is portable and convenient to attach anywhere in the application as designed to redirect the execution without being modified the existing code.

Office Action, page 7. Applicant respectfully disagrees with this assertion and asks for further clarification from the Examiner regarding how Krishnan or Hall render obvious the directory feature recited in claim 14, namely "setting a predetermined directory such that an application running under the predetermined directory will be redirected to the virtual OS environment based on the location of the application being under the predetermined directory."

CONCLUSION

All rejections have been addressed and Applicant has made a diligent effort to place this application in condition for immediate allowance. Accordingly, reconsideration and allowance are respectfully requested and the Examiner is respectfully requested to pass this application to issue. If the Examiner is unable to issue an immediate Notice of Allowance, the Examiner is respectfully requested to telephone Applicants' attorney with a view towards resolving the outstanding issues prior to the issuance of an Advisory Action.

It is believed that any fees associated with the filing of this paper are identified in an accompanying transmittal. However, if any additional fees are required, they may be charged to Deposit Account 50-3583, in the name of AdvantEdge Law Group, LLC. To the extent necessary, a petition for extension of time under 37 C.F.R. 1.136(a) is hereby made, the fee for which should be charged against the aforementioned account.

Respectfully submitted,

Date: December 5, 2007

By: 
L. Eric Gardanier
Attorney for Applicants
Registration No. 40,617
Customer No. 53124